

Documentation technique

Évaluation en cours de formation

Rabillard Fabien | Studi 2022

Projet Hypnos

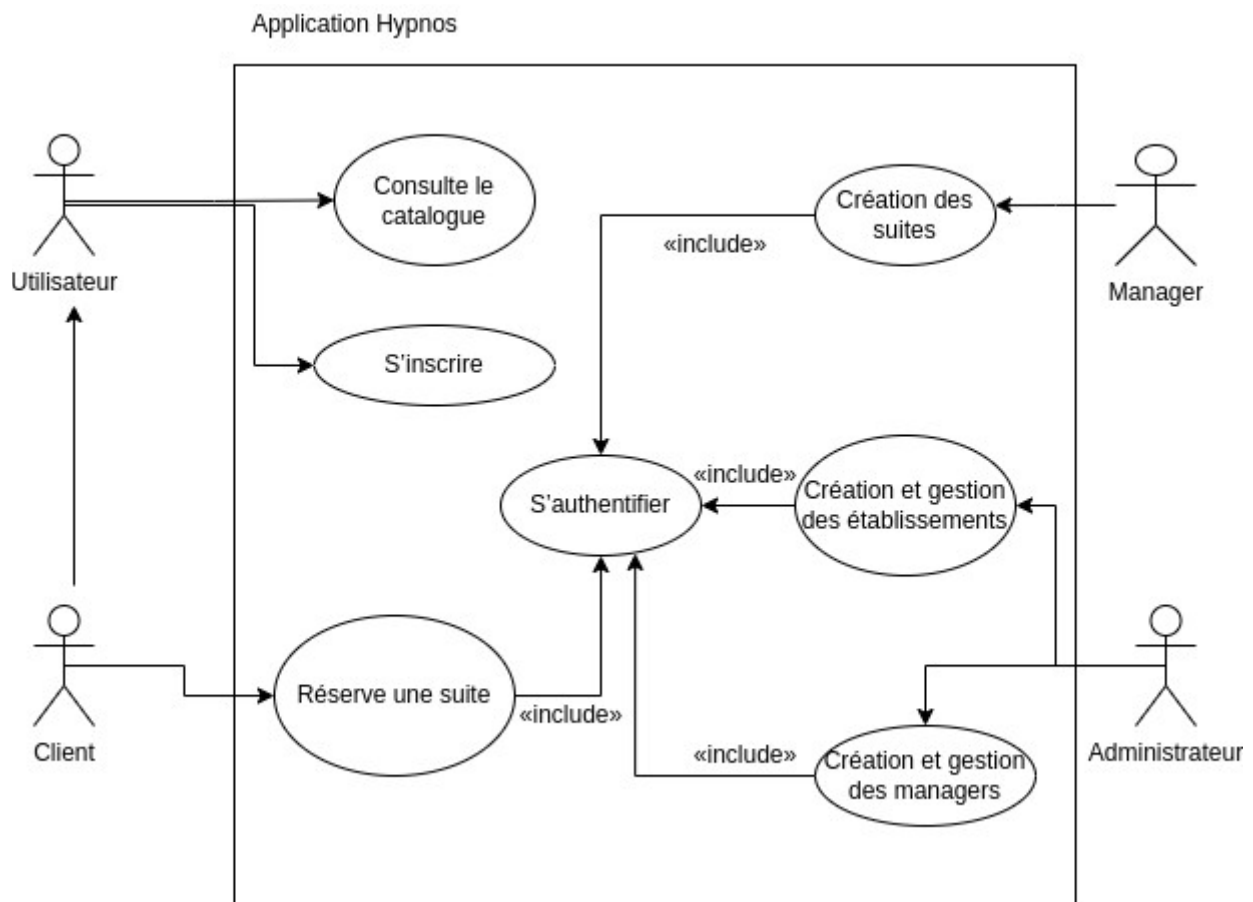
Sommaire

- Spécifications techniques	page 2
- Diagramme de Cas d'utilisation	page 3
- Diagrammes de Séquence	pages 4-5
- Diagramme de Classe	page 6
- Modèle conceptuel de données	page 7
- Script Sql	page 8
- Annexe	page 9

Spécifications techniques

Local	Production
<p>Serveur :</p> <ul style="list-style-type: none">- Ubuntu 21.10- Mysql (version 8.0.28)- PHP 8 (version 8.0.17)- Apache (version 2.4.48)- Node.JS (version 6.13.2) <p>Versioning :</p> <ul style="list-style-type: none">- Git- Github	<p>Serveur :</p> <p>Heroku :</p> <ul style="list-style-type: none">- Mysql via JawsDB MySQL add-on- Apache <p>Via buildpacks</p> <ul style="list-style-type: none">- Node JS- PHP 8 (version PHP 8.0.17)
<p>Front :</p> <ul style="list-style-type: none">- HTML 5- CSS 3- Bootstrap 5- Twig	
<p>Back :</p> <ul style="list-style-type: none">- Composer version 2.2.7- Symfony 6.0.6 :<ul style="list-style-type: none">Bundles :<ul style="list-style-type: none">• Easyadmin• Doctrine• Security• Twig• Validator• VichUploaderBundle• ...	

Diagramme de Cas d'utilisation



Diagrammes de Séquence

Diagramme de séquence - Réserver une suite

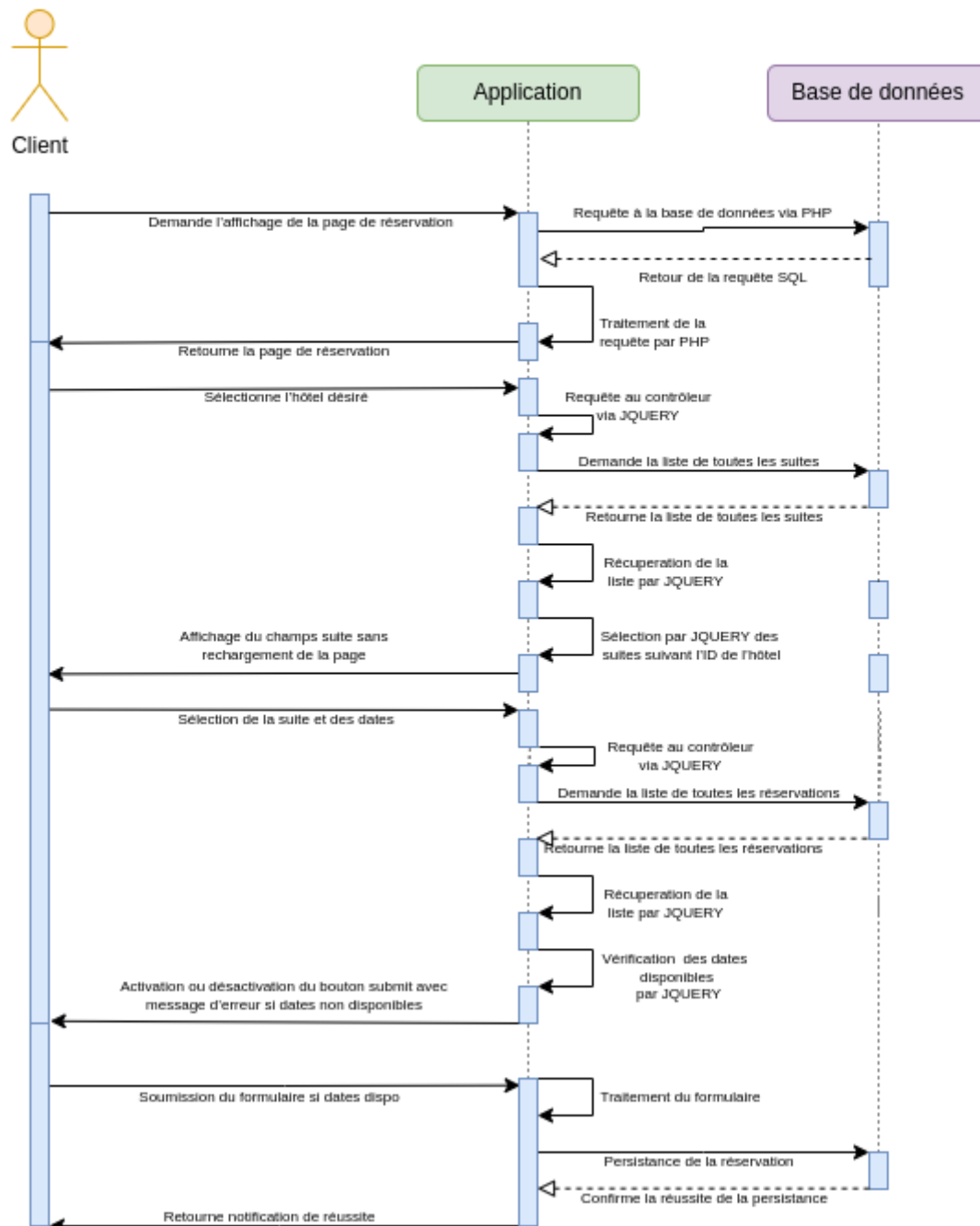


Diagramme de séquence - Consulter ses réservations

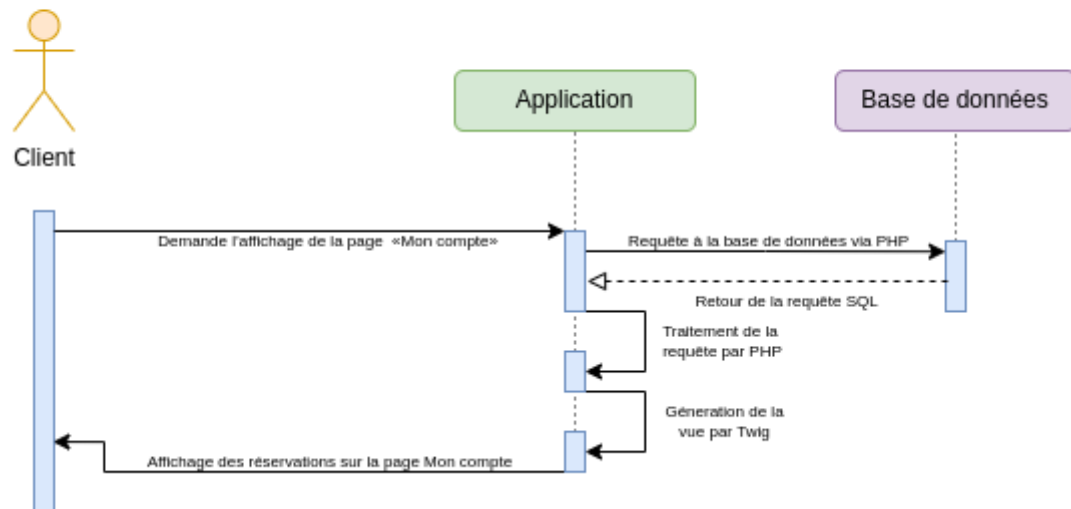
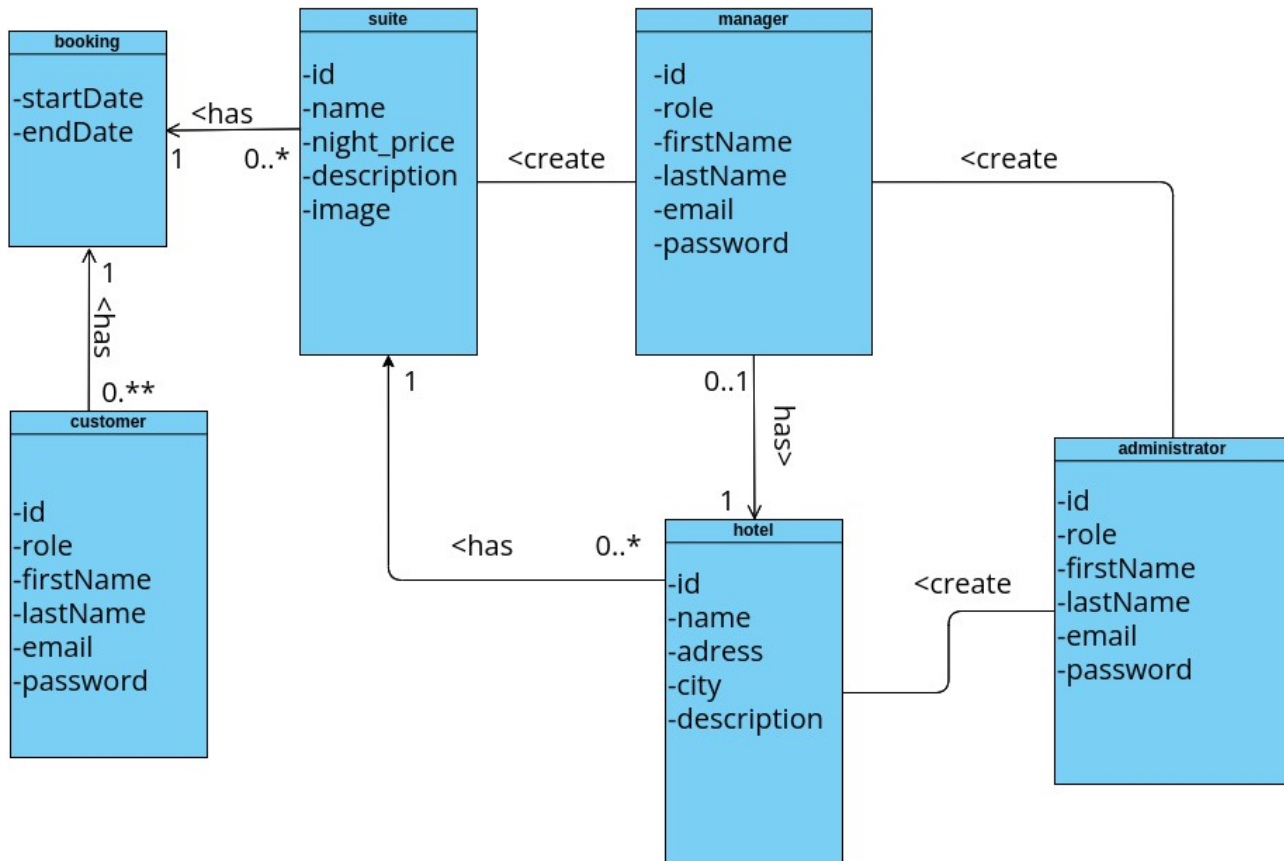
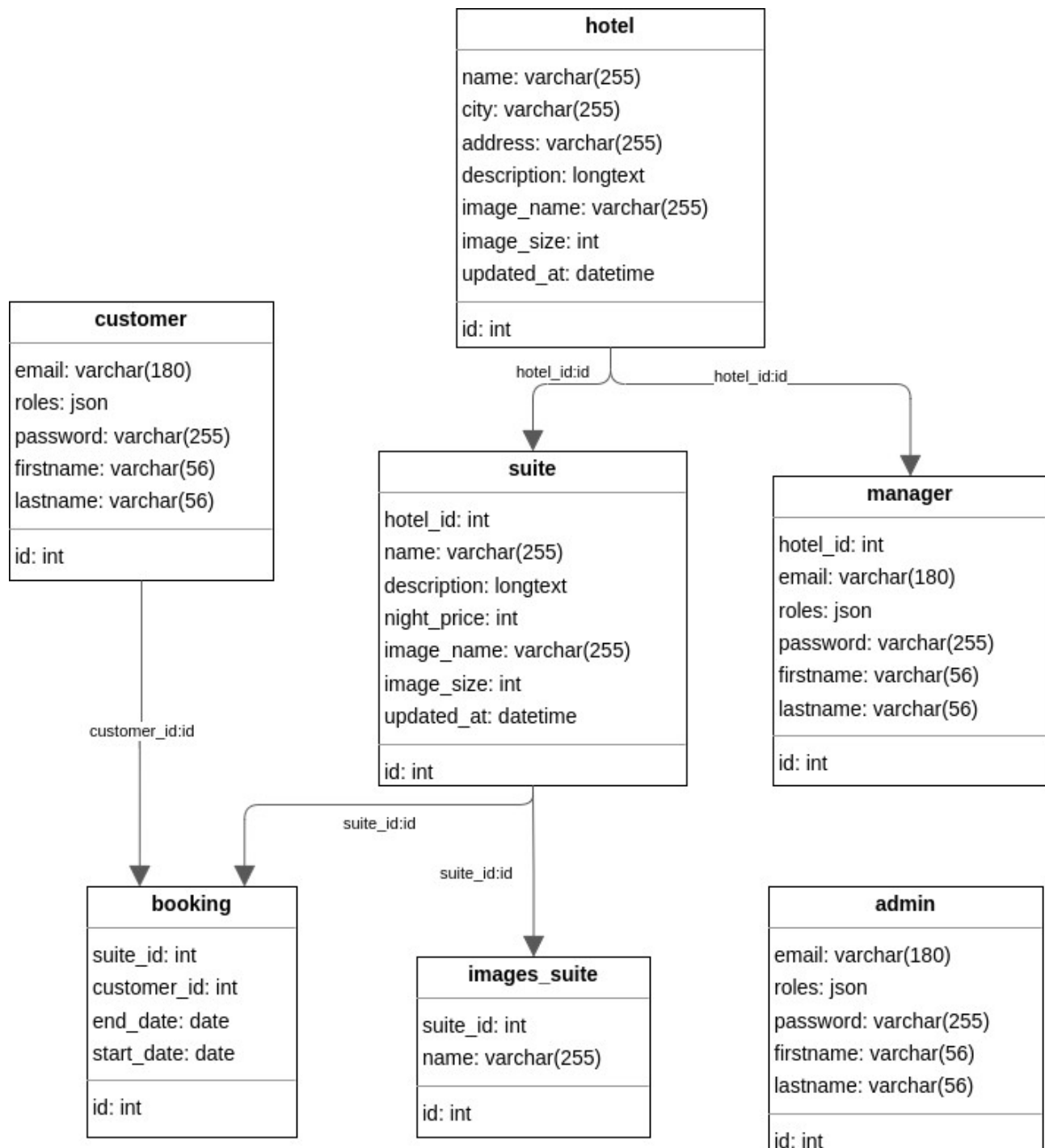


Diagramme de Classe



Modèle conceptuel de données



Script Sql

Avec Symfony, c'est l'ORM Doctrine qui se charge de faire le lien entre notre application et la base de données.

Cela nous permet de rester concentré sur l'application et évite l'écriture de nombreuses requêtes SQL.

La console Symfony (CLI) permet de générer très rapidement les scripts SQL. Pour cela, une fois nos entités créées, il est nécessaire de lancer la commande suivante :

```
php/bin console make:migration
```

Cela génère un fichier de migration comprenant les requêtes SQL à exécuter mais aussi les requêtes inverses (en cas d'erreur, permet d'annuler les requêtes) pour un retour en arrière en sécurité.

C'est toujours bien de contrôler le fichier de migration, cela permet de voir si l'on n'a pas fait d'erreur ou d'oubli :

Par exemple un «NOT NULL» au lieu d'un «NULL».

Dans ce cas il faut retourner dans le fichier de l'entité concernée, faire la modification et relancer la commande `php/bin console make:migration`.

C'est mieux que de modifier le fichier de migration directement, surtout si d'autres personnes travaillent ou reprennent le projet,

Ensuite, il faut exécuter les requêtes SQL afin de les persister en BDD.

La commande est :

```
php bin/console doctrine:migrations:migrate
```

Parfois, il est quand même nécessaire d'écrire une requête SQL comme par exemple pour créer notre administrateur :

```
INSERT INTO admin (id, email, roles, password, firstname, lastname) VALUES  
(NULL, 'admin@test.fr', ['ROLE_ADMIN'],  
'$2y$13$vr1lIFEr35ilxDHeuszJGeNdfivcBaKB8XpcwaoyqYiAOrlJMCony',  
'Administrateur', 'Principal');
```


Annexe

Fichier migration de Symfony

Application Hypnos

```
<?php

declare(strict_types=1);

namespace DoctrineMigrations;

use Doctrine\DBAL\Schema\Schema;
use Doctrine\Migrations\AbstractMigration;

/**
 * Auto-generated Migration: Please modify to your needs!
 */
final class Version20220419215646 extends AbstractMigration
{
    public function getDescription(): string
    {
        return '';
    }

    public function up(Schema $schema): void
    {
        // this up() migration is auto-generated, please modify it to your needs
        $this->addSql('CREATE TABLE `admin` (id INT AUTO_INCREMENT NOT NULL, email VARCHAR(180) NOT NULL, roles JSON NOT NULL, password VARCHAR(255) NOT NULL, firstname VARCHAR(56) NOT NULL, lastname VARCHAR(56) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
        $this->addSql('CREATE TABLE booking (id INT AUTO_INCREMENT NOT NULL, suite_id INT DEFAULT NULL, customer_id INT DEFAULT NULL, end_date DATE NOT NULL, start_date DATE NOT NULL, INDEX IDX_E00CEDDE4FFCB518 (suite_id), INDEX IDX_E00CEDDE9395C3F3 (customer_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
        $this->addSql('CREATE TABLE customer (id INT AUTO_INCREMENT NOT NULL, email VARCHAR(180) NOT NULL, roles JSON NOT NULL, password VARCHAR(255) NOT NULL, firstname VARCHAR(56) NOT NULL, lastname VARCHAR(56) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
        $this->addSql('CREATE TABLE hotel (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, city VARCHAR(255) NOT NULL, address VARCHAR(255)');
    }
}
```

```

NOT NULL, description LONGTEXT NOT NULL, image_name VARCHAR(255) NULL,
image_size INT NULL, updated_at DATETIME NULL, PRIMARY KEY(id)) DEFAULT
CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
$this->addSql('CREATE TABLE images_suite (id INT AUTO_INCREMENT NOT
NULL, suite_id INT DEFAULT NULL, name VARCHAR(255) NOT NULL, INDEX
IDX_3804059A4FFCB518 (suite_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET
utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
$this->addSql('CREATE TABLE manager (id INT AUTO_INCREMENT NOT NULL,
hotel_id INT DEFAULT NULL, email VARCHAR(180) NOT NULL, roles JSON NOT NULL,
password VARCHAR(255) NOT NULL, firstname VARCHAR(56) NOT NULL, lastname
VARCHAR(56) NOT NULL, UNIQUE INDEX UNIQ_FA2425B9E7927C74 (email), UNIQUE
INDEX UNIQ_FA2425B93243BB18 (hotel_id), PRIMARY KEY(id)) DEFAULT CHARACTER
SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE = InnoDB');
$this->addSql('CREATE TABLE suite (id INT AUTO_INCREMENT NOT NULL,
hotel_id INT DEFAULT NULL, name VARCHAR(255) NOT NULL, description LONGTEXT
NOT NULL, night_price INT NOT NULL, image_name VARCHAR(255) NULL,
image_size INT NULL, updated_at DATETIME NULL, INDEX IDX_153CE4263243BB18
(hotel_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE
`utf8mb4_unicode_ci` ENGINE = InnoDB');
$this->addSql('CREATE TABLE messenger_messages (id BIGINT
AUTO_INCREMENT NOT NULL, body LONGTEXT NOT NULL, headers LONGTEXT NOT
NULL, queue_name VARCHAR(190) NOT NULL, created_at DATETIME NOT NULL,
available_at DATETIME NOT NULL, delivered_at DATETIME DEFAULT NULL, INDEX
IDX_75EA56E0FB7336F0 (queue_name), INDEX IDX_75EA56E0E3BD61CE
(available_at), INDEX IDX_75EA56E016BA31DB (delivered_at), PRIMARY KEY(id))
DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4_unicode_ci` ENGINE =
InnoDB');
$this->addSql('ALTER TABLE booking ADD CONSTRAINT
FK_E00CEDDE4FFCB518 FOREIGN KEY (suite_id) REFERENCES suite (id)');
$this->addSql('ALTER TABLE booking ADD CONSTRAINT
FK_E00CEDDE9395C3F3 FOREIGN KEY (customer_id) REFERENCES customer (id)');
$this->addSql('ALTER TABLE images_suite ADD CONSTRAINT
FK_3804059A4FFCB518 FOREIGN KEY (suite_id) REFERENCES suite (id)');
$this->addSql('ALTER TABLE manager ADD CONSTRAINT
FK_FA2425B93243BB18 FOREIGN KEY (hotel_id) REFERENCES hotel (id)');
$this->addSql('ALTER TABLE suite ADD CONSTRAINT FK_153CE4263243BB18
FOREIGN KEY (hotel_id) REFERENCES hotel (id)');
}

public function down(Schema $schema): void
{
    // this down() migration is auto-generated, please modify it to your needs
    $this->addSql('ALTER TABLE booking DROP FOREIGN KEY
FK_E00CEDDE9395C3F3');
    $this->addSql('ALTER TABLE manager DROP FOREIGN KEY
FK_FA2425B93243BB18');
    $this->addSql('ALTER TABLE suite DROP FOREIGN KEY
FK_153CE4263243BB18');
    $this->addSql('ALTER TABLE booking DROP FOREIGN KEY

```

```
FK_E00CEDDE4FFCB518');  
    $this->addSql('ALTER TABLE images_suite DROP FOREIGN KEY  
FK_3804059A4FFCB518');  
    $this->addSql('DROP TABLE `admin`');  
    $this->addSql('DROP TABLE booking');  
    $this->addSql('DROP TABLE customer');  
    $this->addSql('DROP TABLE hotel');  
    $this->addSql('DROP TABLE images_suite');  
    $this->addSql('DROP TABLE manager');  
    $this->addSql('DROP TABLE suite');  
    $this->addSql('DROP TABLE messenger_messages');  
}  
}
```